

拡張ラスタオペレーションによる3次元図形表示と画像操作†

藤田 良^{††} 古賀 和義^{††}
 福永 泰^{††} 西田 健彦^{†††}

計算機システムにおけるユーザインタフェースの中核をなす表示システムにおいては、CADの分野を中心に、2次元図形表示はもちろん、陰影の付いた3次元図形の表示性能の高速化が重要となってきている。また、DTPを代表とする高度な文書処理システムにおいては、種々のサイズの文字を表示したり、画像データの拡大、縮小、回転の操作を十分短い応答時間で実行する性能が必要である。そこで、これらの要求に応えるために、ハードウェアで実現するプリミティブ機能として、拡張ラスタオペレーション機能を提案する。この機能は、従来のラスタオペレーションが矩形の画像間の演算を行う機能であるのに対し、ソース画像を3次元の任意方向の直線上に写像し、同時に始終点間の輝度の補間と、Zバッファによる隠面消去を行えるように拡張した機能である。本機能を繰り返し実行することにより上記表示機能を実現することができる。次に、拡張ラスタオペレーションを実現した実際のハードウェア構成と、3次元図形の表示と画像操作を行うファームウェアについて述べ、拡張ラスタオペレーションの有効性を明らかにする。

1. はじめに

マイクロプロセッサの性能向上に伴って、高速な演算能力と高度なグラフィックス機能をユーザ1人で占有するワークステーションが普及してきた。ワークステーションにおける表示システムはマンマシンインタフェースの中核をなしており、作業効率の向上のためには、高機能化、高速化が重要となってきている¹⁾。

一方、ワークステーションは、CAD (Computer Aided Design) や CAE (Computer Aided Engineering) のほか、DTP (Desk Top Publishing) などで使用されており、その表示システムは、図形、画像、文字の3種類のデータを取り扱う必要がある。

図形データは、直線、円、多角形などのように幾何データであり、その中で、より高機能な図形データとしては、3次元空間上に定義された物体に陰影を付けて表示するための、3次元輝度付き平面がある²⁾。

また、画素の集合として与えられる画像データの処理としては、ラスタオペレーションやセルアレイ³⁾などの機能が必要である。ラスタオペレーションは、2つの画像間の論理演算を行う機能で、カーソルの表示や、ウィンドウのスクロール、移動などに用いられ、とくに高速性が必要な機能である。セルアレイは、画像データの拡大、縮小、回転などを一般化した機能で

あり、GKS³⁾、PHIGS⁴⁾などのグラフィックインタフェースにおいて標準の出力プリミティブとして定義されている。

文字データには、直線列で文字を表すストロークフォント、多角形のアウトラインフォントと、2値の画像データであるドットフォントの3種類がある。フレームメモリに展開する時点では、前者2つは、図形として扱うことが可能であり、後者は、画像として扱えるため、文字データに関しては、図形と画像のいずれかの機能で対応することができる。

従来のハードウェアは、これらの機能を高速に行う処理系を個別に最適化したアーキテクチャを実現していた^{5)~7)}。しかし、こうした図形、文字、画像の処理をさらに詳細に考察した結果、多数の線形補間演算の繰返して実現できるという共通性のあることがわかった。例えば、3次元輝度付き平面は、X, Y, Z座標および赤、緑、青(以下R, G, B)の輝度レベルについての線形補間を行うことで実現可能である。また、セルアレイは、さらに、ソースとなる画像データの参照座標の線形補間を行うことで実現できる。

そこで、上記の機能を統合したハードウェアプリミティブとして、これらX, Y, Z座標、R, G, Bの輝度、ソースのX座標の7種の線形補間を基本とする拡張ラスタオペレーションを提案する。そして、この拡張ラスタオペレーションを複数個の同一品種のLSIによって並列処理するハードウェアの構成を示し、その有効性を明らかにする。

† Three Dimensional Graphics and Image Transfer Using Extended Raster Operation by RYO FUJITA, KAZUYOSHI KOGA, YASUSHI FUKUNAGA (Hitachi Research Laboratory, Hitachi, Ltd.) and TAKEHIKO NISHIDA (Omika Works, Hitachi, Ltd.).

†† (株)日立製作所日立研究所

††† (株)日立製作所大みか工場

2. 拡張ラスタオペレーション

2.1 ラスタオペレーション²⁾

拡張ラスタオペレーションについて述べる前に、従来のラスタオペレーションの概要を示す。

ラスタオペレーションは、カーソルの表示や、ウィンドウのスクロール、移動の処理をハードウェアで高速に行う機能であり、式(1)に示すように矩形領域間の画像データの演算として定義される。

$$D' = F(S, D) \quad (1)$$

ここで、 S は、ソースの矩形画像データ、 D は、デスティネーションの矩形画像データである。関数 F は、一般に論理関数として与えられる。 D' は、 D の場所に新たに展開される画像データである。図1にラスタオペレーションの概念図を示す。ラスタオペレーションは、関数 F を必要に応じて設定することにより種々の操作、例えば画面のクリアや移動、文字の展開などを実現することができる。しかし、このラスタオペレーションのみで、他の図形描画である直線発生や多角形塗り潰し等を行おうとすると、十分な応答性を得ることができない。また、3次元図形を表示するときの隠面除去や輝度の補間を行う処理、そして画像の拡大、縮小等の処理まで考えると、ラスタオペレーションの機能だけでは困難で、専用のハードウェアを設け高速化を図る必要がある。

2.2 3次元図形の表示方法

エンジニアリングワークステーションに必要な表示機能の中で、最も複雑な処理を必要とするのが3次元図形の表示機能である。

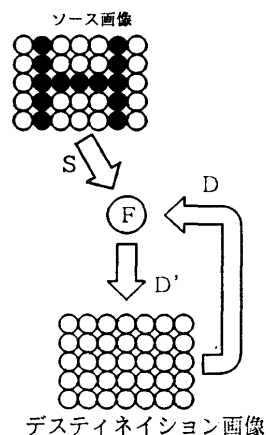


図1 ラスタオペレーションの概要

Fig. 1 Raster operation that executes logical operation between two images.

3次元空間上の立体図形に陰影を付けてリアルにディスプレイ上に表示するための方法としては、グーロー (Gouraud) シェイディングやフォン (Phong) シェイディングなどがある²⁾。

グーローシェイディングでは、図2に示すように、立体図形を多数の小平面に分割して定義し、平面の各頂点には、法線 $N_0 \sim N_3$ を与える。この法線と、光源および立体の材質などから各頂点の輝度 $I_0 \sim I_3$ を求め、さらに、平面内の各画素の輝度を線形補間することにより、滑らかな陰影を付けることができる。平面の内部の画素の輝度 I_p は、まず左辺と右辺の輝度 I_L と I_r を頂点輝度 $I_0 \sim I_3$ から補間して求め、さらに I_L と I_r の間を補間して求める。各画素の輝度 I_p は、光の三原色の R, G, B を要素とするベクトルで、それぞれ独立に求めることが可能である。また各画素の Z 座標も同様に、周囲の4頂点の値から線形補間して求めることができる。

一方、フォンシェイディングは、グーローシェイディングが輝度を補間しているのに対し、法線を補間する。すなわち、3次元平面の各頂点の法線から、平面内部の各画素に対する法線ベクトルを補間して求め、1画素ごとに輝度の算出を行う。

このようにして、平面内部の全画素の輝度および Z 座標を求め、これをフレームメモリに隠面除去を行いながら展開することによって、滑らかな陰影の付いた立体図形の表示を行うことができる。

2.3 セルアレイ

画像の拡大、縮小、回転を行う画像操作のプリミティブであるセルアレイは、通常図3のように矩形のソース画像を、任意の四角形に写像する機能として定義されている。PHIGS の場合には、さらに、画像の領域を示す4頂点に Z 座標が与えられており、 Z 座標の補間も必要である。セルアレイを実現する手段としては、定式化された方法がないため、ここでは図3のようにソース画像の水平な画素列 (P_0, P_1) をデスティ

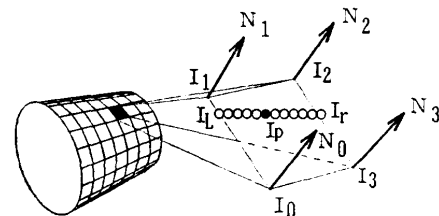


図2 3次元図形の表示方法

Fig. 2 Rendering method of shaded polygons which are subdivided from a solid surface.

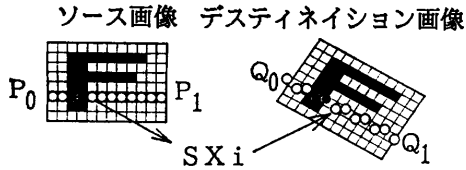


図3 セルアレイの概要

Fig. 3 Rendering method of CELL_ARRAY function that transforms image data.

ネーション上の任意直線 (Q_0, Q_1) に写像する方式を用いることにした。この方式は、ソース画像をラスタ方向に順次アクセスし、デスティネーション画像については、通常の直線展開と同じアドレスを生成することによって実現できる。具体的には、デスティネーションとなる直線上の画素に対応する参照座標 SX_i を線形補間によって求め、ソース画像から画素データを読み出し、デスティネーションに書き込む処理を繰り返すことによって実現できる。しかし、この方式を単純に行うと、隣合う直線間に隙間ができ、画素の抜けが生じてしまう。そこで、次のような方法で解決した。

まず、デスティネーションの直線は、Bresenham のアルゴリズム⁸⁾に従って8連結の画素列を発生させる。そして、その始点 Q_0 と終点 Q_1 は、4連結の画素列をスキャンさせるようにした。こうすることによって、8連結の直線の画素密度 (単位長さあたりの画素数) が、 $(1/\sqrt{2}) \sim 1$ と疎であるのに対し、4連結の画素密度は、 $1 \sim \sqrt{2}$ と密であるため、画素の抜けを発生しないようにすることができる。

2.4 拡張ラスタオペレーション

前節までに示したように、3次元図形は、水平な直線上の画素の輝度 R, G, B と Z 座標を補間する機能を必要とし、セルアレイでは、ソース画像の参照座標 SX 上の画素を任意の直線上の画素に転送する機能を必要としている。他の2次元、3次元の直線や、文字などもこれらのサブセットとなっている。そこで、これらの処理を、統合化した機能として、拡張ラスタオペレーションを以下のように定義した。

拡張ラスタオペレーションは、図4に示すように、 m 画素の水平に並ぶソース画像を、任意の方向成分を有する n 画素のデスティネーション画像へ写像することを基本とする。ここで画素 P_i は、デスティネーションの画素の座標 X_i, Y_i 、画素の深さを示す Z_i 、画素の色を与える光の三原色 R, G, B の各成分 R_i, G_i, B_i 、ソース画像の参照座標 SX_i を要素とする7次元のベクトルであり、式(2)のように表す。

$$P_i = (X_i, Y_i, Z_i, R_i, G_i, B_i, SX_i) \quad (2)$$

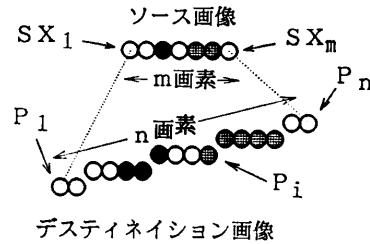


図4 拡張ラスタオペレーション

Fig. 4 Extended raster operation.

$$P_{i+1} = P_i + \Delta P \quad (3)$$

$$\therefore \Delta P = \frac{P_n - P_1}{n}$$

したがって、デスティネーション画像の始点、終点の画素 P_1, P_n に対して、その中間の画素 P_i ($i=2, \dots, n-1$) は式(3)で示したような線形補間によって求めることができる。

また、拡張ラスタオペレーションは、さまざまな出力プリミティブを実行させるために、さらに、以下に示す機能を付加した。

(1) ソース画像の写像倍率モード

ソース画像の参照座標 SX を算出方法として2つのモードを用意した。

1つは、破線や、ハッチングによる塗り潰しを行うためのモードで、ソース画像を繰り返しサイクリックに参照して、2倍、4倍等の整数倍率でデスティネーション画像に写像する機能である。

もう1つは、セルアレイなどで用いる任意倍率の写像を行うモードで、図4に示したように m 画素のソース画像を n/m 倍し、 n 画素のデスティネーションに写像する機能である。

(2) 2値画像のカラー化

拡張ラスタオペレーションは、基本的には、多値のフルカラー (1600 万色) 画像を取り扱う機能である。したがって文字フォントのような2値画像は、多値の画像に変換し、カラー化を行う必要がある。そこで、2値のソース画像は、 SX_i で参照するソース画像の値が“1”の画素は、式(3)で順次求める前景色 R_i, G_i, B_i に変換する。また、“0”の画素は、あらかじめ設定した背景色の色とするか、 P_i を発生しないようにした。

(3) 隠面消去機能

3次元の図形を表示するときは、重なりのある図形のうち、手前にあるものを優先して描画する隠面消去を行う必要がある。ここでは、 Z バッファを用いる隠面消去の方式を採用した。すなわち、式(3)で順次求

められる Z_i に対して、既に Z バッファ上に格納されている Z の値と比較を行い、 Z_i が大きい(手前にある)時に画素 P_i を発生するようにした。

以上示したように、拡張ラスタオペレーションは、3次元空間上の直線を展開するプリミティブであり、繰り返し実行させることにより、従来のラスタオペレーションを始め3次元の平面や、セルアレイの展開などのさまざまな出力プリミティブを実現することが可能となる。

3. ハードウェア構成

3.1 システム構成

拡張ラスタオペレーションの有効性を明らかにするため、図5に示すような表示システムを試作した。この表示システムは、コマンド解釈部、レンダリング処理部、表示部の3ユニットから構成しており、それぞれ以下のような処理を行う。

(1) コマンド解釈部

32bits の汎用マイクロプロセッサである MPU は、上位プロセッサからのコマンドを FIFO (First-In First-Out buffer) レジスタを介して受け取り、RAM に格納されているプログラムに従って、レンダリング部に対して拡張ラスタオペレーションを起動する。

コマンドとしては、PHIGS, PHIGS+⁹⁾ に対応できる直線、文字、3次元平面などの出力プリミティブを用意した。また、さらにプログラムを変更すること

によって、コマンドの追加やインタフェースの変更を行うことが可能である。

(2) レンダリング処理部

レンダリング処理部は、拡張ラスタオペレーションを実行する LSI, RCPE (Rendering Control Processor Element) を複数個用いて構成している。RCPE は、コマンド解釈部からの起動に従って、表示部のフレームメモリおよび Z バッファに対して並列に描画処理を行うことができる。

(3) 表示部

表示部は、1600万色のフルカラー表示を行わせるため R, G, B 各 8 bits の階調を保持する 24 プレーンのフレームメモリと、3次元図形の隠面消去を行うため 16 bits の Z 座標を保持する 16 プレーンの Z バッファを有している。さらに、カーソルやプリンク等の表示を行わせるために、4 プレーンの画像メモリを設け、トータル 44 プレーンのフレームメモリを構成した。

CRTC (CRT Controller) は、フレームメモリを常時読み出し、DAC (Digital to Analogue Converter) でビデオ信号に変換して、1,280×1,024画素の精細度を持つ CRT (Cathode Ray Tube) に表示を行う。

さて、図4に示した表示システムは、上位プロセッサとコマンド解釈部を FIFO を介して接続しているため、パイプラインに処理を実行させることができる。また、レンダリング部では、44 プレーンのフレームメモリを並列にアクセスしているため、高速性能を得ることが可能である。

3.2 RCPE の構成

拡張ラスタオペレーションは、式(2)に示す7個の要素に対する補間演算器 DDA (Digital Differential Analyzer)²⁾ が必要である。各 DDA は、式(3)に示したベクトル演算を行うため、各要素を独立に処理することができる。そこで、レンダリング処理部を4プレーン単位に分割し、複数の LSI から構成した。これによって、フレームメモリとのバス幅を増し、高速なアクセスができるようになる。今回試作した RCPE は、各プレーンごとに 16 bits, 16画素分のデータバスを有し、最大11個の RCPE を並列に動作させることで、55 M 画素/s, 300 MB/s 以上のスループットを実現している。

しかし、レンダリング部をそのまま4プレーン単位に分割したのでは、11種類の LSI を開発しなくてはならず効率がよくない。そこで、拡張ラスタオペレー

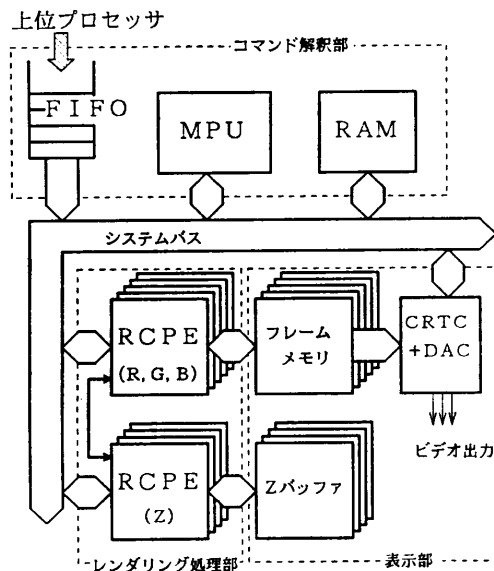


図5 表示システムの構成

Fig. 5 Rendering system that based on pipeline and parallel architecture.

ションの行う演算の独立性と、分割した LSI の共通性に着目し、RCPE は、外部から与えられる識別信号 ID に対応して処理を変えられるようにした。これによって、レンダリングプロセッサを 1 品種の LSI で構成することができる。また、Z バッファが不要なシステムや、色数の少ないシステムに対しても、効率良く対応できる柔軟性のある構成とすることができた。

RCPE の内部は、図 6 に示すように 4 種類の DDA と、アドレス制御部、データ制御部、シーケンス制御部から構成されている。

DDA には、各プレーンに共通な X, Y, SX と、 R, G, B, Z のどれか 1 つの動作を、識別信号 ID に従って実行する合計 4 個の DDA があり、RCPE ごとに、式(3)に示した補間演算を並列に実行する。

アドレス制御部は、 X, Y, SX の DDA からの出力をフレームメモリまたは Z バッファのアドレスに変換する回路である。

データ制御部は、ソース画像からフレームメモリへの書き込みデータを作成し、式(1)に示したフレームメモリ間の論理演算と、隠面除去のために Z 座標の比較演算を行う。ソース画像としては、①フレームメモリから読み出した画像データ、②MPU が直接与える多値の画像データ、③MPU が直接与える 2 値パターンを DDA (ID) の画素データでカラー拡張した画像データ、のうち 1 つを選択することができる。

制御回路は、MPU が与えるコマンドに従って RCPE を制御するユニットである。

このように R, G, B, Z の独立性と、プレーン単位

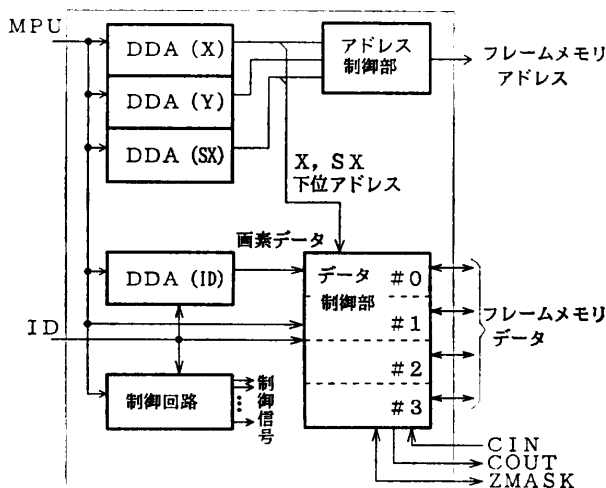


図 6 RCPE の構成

Fig. 6 Structure of the rendering control processor element (RCPE).

の独立性を利用して、拡張ラスタオペレーションを複数の LSI に分割して並列処理を行わせた。しかし、RCPE は、4 プレーンすなわち 4 bits 分のデータしか扱えないため、隠面除去を行うためには、Z 比較計算のポロー信号を RCPE 間で伝搬させる必要がある。そこで、下位の RCPE からのポローの入力 CIN と、上位へのポロー出力 COUT を設け、RCPE 間をポローが伝搬できるようにした。最上位からのポロー出力は Z 座標の比較結果となり、各 RCPE の ZMASK 端子によってこれを伝え、隠れる画素の書き込みを抑制できるようにした。

4. 評価

4.1 機能

拡張ラスタオペレーションは、さまざまな出力プリミティブに適用可能で、上位のプロセッサの負荷を軽減させることのできる高機能なプリミティブである。これを明らかにするため、グラフィックス標準仕様である PHIGS と、PHIGS+ の出力プリミティブに対して、拡張ラスタオペレーションがどのように適用され、そしてレンダリング処理部における 7 個の DDA が、どのように動作するかを表 1 に示した。表 1 において、○印は DDA が始点と終点の値を補間し、△印は始点と終点の値が同じで一定の値を出力することを示している。また、×印はその値が出力に影響しないことを示す。

PHIGS と PHIGS+ には、合わせて 17 種類の出力プリミティブがある。これらの機能は、ポリライン、ポリマーカ、テキスト、塗り潰し、セルアレイ、3 次元平面の 6 種類に分類することが可能で、それぞれ以下のように拡張ラスタオペレーションを適用することができる。

(1) ポリライン

ポリラインは、実線、破線といった線種を与えることができる直線列である。この線種を表す 2 値パターンをソース画像とし、整数倍率で写像するように拡張ラスタオペレーションを起動することによって、ポリラインを実現することができる。

(2) ポリマーカ

ポリマーカは、2 値の画像データであるマーカをフレームメモリに描画するプリミティブである。したがって、従来のラスタオペレーションで実現できる。

(3) テキスト

文字列であるテキストには、String, Stroke などの属性を与えることができる。前者は、文字の大きさ、位置などに制限がある文字列で、ドットフォント文字による展開がこれにあたる。後者の Stroke 属性は、大きさ、位置、方向が任意の文字列で、直線で文字を表すストロークフォント文字や、多角形で文字を表すアウトラインフォント文字がある。ドットフォント文字は、ポリマーカと同じように従来のラスタオペレーションで実現できる。またストロークフォント文字は、拡張ラスタオペレーションで直線列を描画し、アウトラインフォント文字は、次の項で示す多角形の塗り潰し機能を用いて実現できる。

(4) 塗り潰し

多角形の内部を塗り潰す出力プリミティブであり、2.2節のように多角形の各辺の情報から、辺上の画素の X, Y, Z 座標を求め、辺の間を水平な直線で描画することにより実現できる。また、塗り潰しは、ハッチ、パターンといった属性を与えることができるが、ハッチの場合には2値、パターンの場合には多値の画像データを、拡張ラスタオペレーションのソース画像として与えることにより対応できる。

(5) セルアレイ

セルアレイは、2.3節に示したように、任意倍率で写像するモードで拡張ラスタオペレーションを、繰り返し実行させて実現できる。

(6) 3次元平面

3次元平面をグーローシェイディングによって表示させるには、2.2節のように、左右辺の間を補間した水平な直線を発生することにより実現できる。一方、フォンシェイディングの場合には、1画素ごとに法線ベクトルの補間と、輝度の算出が必要で、拡張ラスタオペレーションにおける輝度の補間機能を利用することができない。しかし、法線ベクトルの補間演算と、輝度の算出は、MPUによって実行させ、RCPEは、求めた輝度データを多値のソース画像データとした拡張ラスタオペレーションを行うことにより実現できる。

PHIGS+には、上記6種類の機能のほかに、曲線、曲面などの出力プリミティブがある。これらに関しては、それぞれ、十分小さな直線と平面に分割することによって、拡張ラスタオペレーションで対応させることが可能となる。

4.2 性能

今回試作した表示システムにおける、拡張ラスタオペレーションの性能について示す。

拡張ラスタオペレーションを実現するRCPEは、72ns周期のクロックで動作しており、図6に示した各DDAは、1クロックで1画素を生成できる。一方、フレームメモリは、水平な16画素のデータを4クロック、288nsでアクセスすることができる。DDAとフレームメモリのアクセスはパイプラインに実行しているため、拡張ラスタオペレーションの性能は、これ

表1 PHIGSにおける拡張ラスタオペレーションの役割
Table 1 Roles of extended raster operation for output primitives of graphics standards, PHIGS and PHIGS+.

機能	拡張ラスタオペレーションの役割	DDAの動作					
		X	Y	Z	SX	RGB	
PHIGS	ポリライン	1直線の描画	○	○	○	○	△
	ポリマーカ	1ラスタの描画	○	△	△	○	△
	テキスト (String) (Stroke)	1ラスタの描画	○	△	△	○	△
		1直線の描画	○	○	△	△	△
	塗り潰し	多角形の左辺と右辺の間を水平線で描画	○	△	○	○	△
セルアレイ	1ラスタのソース画像を直線上に写像	○	△	○	○	△	
PHIGS+	3次元平面 (Gouraud) (Phong)	多角形の左辺と右辺の間を輝度を補間した水平線で描画	○	△	○	○	○
		多角形の左辺と右辺の間にソース画像を写像	○	△	○	○	×

○: 補間, △: 一定, ×: 未使用.

らの遅い方で決まる。

例えば、水平な直線を描画する場合、RCPEは72 ns/画素で画素を生成する。一方、フレームメモリは、Zバッファの読み出しと、画素の書き込みのために2回のアクセスを行い、36 ns/画素(288 ns×2回/16画素)で処理が行える。したがって、水平な直線の性能は、72 ns/画素となる。これが垂直な直線の場合には、フレームメモリを1回に1画素しかアクセスできないため、576 ns/画素(288 ns×2回/1画素)の性能となる。

セルアレイのようにソース画像が必要な場合には、ソース画像のリード時間18 ns(288 ns/16画素)を上記の性能に加えることによって、性能を求めることができる。

RCPEの性能は、MPU等のオーバヘッドを無視できるような条件で実測した結果、上記で求められる性能と一致した。表示システムとしての性能は、MPUの演算時間が影響するため、十分高速なMPUを採用する必要がある。

5. あとがき

3次元図形や画像の操作を統合化して処理するための拡張ラスタオペレーションは、ソースとなる画像データを、3次元上の直線に写像し、同時に始点、終点に与えられたR, G, Bの輝度データを線形補間する高機能なプリミティブである。現在、高度なグラフィックス標準仕様として知られているPHIGSやPHIGS+における出力プリミティブを拡張ラスタオペレーションの繰返しで実現することができる。さらに、拡張ラスタオペレーションを行うハードウェアは、座標や、輝度の補間演算が独立であるという並列性を利用して、複数のLSIによって並列に処理するように構成し、高速な描画性能を得られるようにした。そのLSIは、識別信号IDに従って、対応する動作を行うようにしたことで、1品種のLSIとすることができた。試作した表示システムにおける、拡張ラスタオペレーションの性能は、72 nsであり、1画面分の1,280×1,024画素を1秒間に10回書き換えられる性能を達成している。

謝辞 本研究に関して、貴重なご意見を頂いた(株)日立製作所マイクロエレクトロニクス機器開発研究所の金間誠一主任研究員に深く感謝いたします。

参 考 文 献

- 1) Thadhani, A. J.: Interactive User Productiv-

ity, *IBM Syst. J.*, Vol. 20, No. 4, pp. 407-423 (1981).

- 2) Newman, W. M. and Sproull, R. F.: *Principles of Interactive Computer Graphics*, McGraw-Hill (1979).
- 3) ANSI: Special GKS Issue, *Computer Graphics*, A Quarterly Report of ACM SIGGRAPH (1984).
- 4) International Standard: Information Processing Systems—Computer Graphics—Programmer's Hierarchical Interactive Graphics System (PHIGS)—, ISO/IEC 9592-1 (1989).
- 5) Torborg, J. G.: A Parallel Processor Architecture for Graphics Arithmetic Operations, *Comput. Gr.*, Vol. 21, No. 4, pp. 197-204 (1987).
- 6) Swanson, R. W. and Thayer, L. J.: A Fast Shaded-Polygon Renderer, *Comput. Gr.*, Vol. 20, No. 4, pp. 95-100 (1986).
- 7) 宮沢, 田中, 甲斐, 永松, 朝日, 鷺見: 高速データ転送機能を備えたグラフィックプロセッサ, 電子情報通信学会集積回路研究会資料, ICD 88-11, pp. 77-82 (1988).
- 8) Bresenham, J. E.: Algorithm for Computer Control of a Digital Plotter, *IBM Syst. J.*, Vol. 4, No. 1, pp. 25-30 (1965).
- 9) Dam, A. V.: PHIGS+ Functional Description Revision 3.0, *Comput. Gr.*, Vol. 22, No. 3, pp. 125-218 (1988).

(平成元年6月29日受付)

(平成2年6月4日採録)



藤田 良

昭和33年生。昭和55年電気通信大学応用電子工学科卒業。昭和57年東北大学大学院修士課程修了。同年(株)日立製作所入社。現在日立研究所においてワークステーションの表示システムに関する研究に従事。電子情報通信学会会員。



古賀 和義 (正会員)

昭和33年生。昭和56年九州大学工学部電気工学科卒業。昭和58年同大学院工学研究科電子工学専攻修了。同年(株)日立製作所入社。現在日立研究所において表示システムに関する研究に従事。マン・マシンインタフェースに興味を持つ。電子情報通信学会会員。

**福永 泰 (正会員)**

昭和25年生。昭和50年京都大学大学院修士課程電気系工学研究科修了。同年(株)日立製作所入社。現在同社日立研究所研究員。計算機アーキテクチャ、グラフィックス、マン・マシンインタフェースの研究に従事。電子情報通信学会、ACM 各会員。

**西田 健彦 (正会員)**

昭和26年生。昭和49年東京大学工学部電気工学科卒業。同年(株)日立製作所入社。現在同社大みか工場にてEWSのハードウェア開発に従事。昭和58年米国UCLAコンピュータサイエンス科修士修了。現在同工場情報端末設計部主任技師。